REMARKS

I. INTRODUCTION

Applicant is filing this request for reconsideration at the suggestion of Examiner Kakali Chaki. Applicant wishes to thank Examiner Chaki for her suggestion and consideration in conducting a short telephone interview with applicant's attorney on September 21, 2005. As discussed during the telephone interview, applicant's attorney previously conducted an interview with Examiner Tuan A Vu, which did not result in the withdrawal of final rejections of the claims based on U.S. Patent No. 6,115,809 to Mattson, Jr., *et al.* either alone or in combination with U.S. Patent No. 5,765,037 to Morrison *et al.* For the reasons briefly explained below, it is clear that these rejections are improper. Applicant is therefore filing this request for reconsideration to avoid filing an appeal, which would needlessly consume the PTO's time and applicant's financial resources.

II. THE SUBJECT MATTER OF THIS APPLICATION

The subject matter of this application is a computer system having a main memory and a cache memory. As is well-known in the art, cache memories are typically relatively high speed memories, such as SRAM, that are used to store frequently accessed data or instructions to avoid the delays inherent in accessing the data or instructions from slower speed main memories, which are generally DRAM. Applicant's computer system executes a program containing information, *i.e.* instructions or data, about which cacheability determinations were made at the time the computer program was compiled. During execution of the program, a processor in the computer system obtains the information from the main memory. The processor then directs selected portions of the information to the cache based at least in part on the cacheability determinations that were made during compilation of the computer program.

In an effort to make this request as concise as possible, applicant will limit his comments to independent claim 1, which reads as follows:

> 1. (Original) A computer system having cache circuitry, the computer system adapted to be controlled by a computer program to cache information, comprising:

cache circuitry, including a cache memory adapted to store information related to a computer program;

a main memory adapted to store the information;

a processor adapted to be controlled by the computer program and adapted to cooperate with a bus interface unit to direct selected portions of the information to the cache circuitry based at least in part on cacheability determinations made during compilation of the computer program; and

bus circuitry, operatively connecting the processor, the cache circuitry, and the main memory.

To support an obviousness rejection of claim 1, the Mattson, Jr. *et al.* patent in combination with the Morrison *et al.* patent would have to disclose a computer system that stores in cache memory portions of the information stored in the main memory based on cacheability determinations that were made during compilation of a computer program being executed by the computer system.

III. DESCRIPTION OF THE PRIOR ART

A. The Mattson, Jr. et al. Patent

The Mattson, Jr. *et al.* patent discloses a method of controlling the manner in which branch instructions are predicted. As is well-known in the art, branch prediction is desirable in a computer system that prefetches instructions for execution. Branch prediction is needed because, when a branch instruction is encountered, the computer system does not know which branch should be followed for fetching additional instructions. In a branch prediction, a processor predicts which branch will be taken and then prefetches the instructions in that branch. In the disclosed method, the branching behaviors of branch instructions in a computer program are first determined. Branch instructions are then partitioned into groups of branch instructions that can be statically predicted and groups of branch instructions that can be dynamically predicted. As described in the Abstract, "dynamic and static code caches are defined in physical memory by allocating pools of memory pages that have prediction flag set to dynamic and static respectively." The blocks of instructions associated with each of the branch instructions are then

stored in the same page of physical memory. A processor fetches and then executes instructions from the same page of physical memory. Thus, all of the instructions are fetched from a single memory, albeit from different pages of that memory. The patent does not disclose fetching and then executing some instructions from a system memory and other some instructions from a cache memory.

The Mattson, Jr. *et al.* patent also fails to disclose a system that makes cacheability determinations, as opposed to branch prediction determinations. As the Examiner may appreciate, the concept of making branch prediction determinations is markedly different from the concept of making cache determinations. Some of the differences between a cacheability determination and a branch prediction determination are:

- a branch prediction applies only to instructions, but a cache determination can apply to data as well as instructions.
- branch prediction is only useful for a program that has branches whereas caching is useful for any program, including those that do not contain branch instructions.
- in branch prediction, all of the instructions are fetched from the same memory whereas, in caching, cached instructions and data are fetched from a memory that is different from the memory from which non-cached instructions and data are fetched.
- branch prediction is only useful in computer systems that prefetch instructions whereas caching is useful in any computer system in which data or instructions are fetched from system memory.

It is therefore clear that the Mattson, Jr. *et al.* patent does not disclose making cacheability determinations during compilation, nor does it disclose storing in cache memory selected portions of data or instructions stored in the main memory of a computer system.

B. The Morrison et al. Patent

The Morrison *et al.* patent describes a technique for more efficiently executing branch instructions in a parallel processing computer system by re-ordering the instructions. The Morrison *et al.* patent teaches analyzing the instructions in a program during compilation based on the manner in which the instruction uses resources, such as whether the instructions are

independent and can therefore be executed in parallel. Each instruction is then assigned an execution time, *i.e.*, the amount of time required to execute the instruction. This execution time is referred to as the "instruction firing time" or "IFT." The instruction is also assigned other data, such as the identity of one of several processors that will execute the instruction. Finally, execution sets of instructions are built by re-ordering the instructions based upon the instruction firing time, the identity of the processor executing the instruction, etc.

The Office Action cites the patent to Morrison *et al.* for disclosing "a system to route instructions data to the appropriate hardware, e.g., cache, using compilation information (see Morrison: col. 33, lines 16-23)." The portion of Morrison cited in the above quote reads as follows:

> The additional and required information comprises two components, a static and a dynamic component; and the information is termed 'shared context storage mapping' (SCSM). The static information results form the compiler output and the TOLL software gleans the information from the compiler generated instruction stream and attaches the register information to the instruction prior to its being received by an LRD.

However, the "additional and required information" referred to has nothing to do with the cacheabililty of information stored in main memory. Instead, as described in the preceding paragraph, "[a]s several users may be executing in the processor elements at the same time, *additional pieces of information* must be attached to each instruction prior to its execution to uniquely identify the instruction source and any resources that it may use." [Col. 33, lines 6-10, emphasis added]. Thus, the "additional and required information" attached to each instruction serves the purpose of identifying the program containing that instruction since several different programs may be executed simultaneously in the disclosed multi-processor system. Contrary to what is implied in the Office Action, the "additional and required information" does not even remotely relate to the cacheability of instructions.

In contrast to storing information in cache memory based on cacheability determinations made at compilation as in applicant's system, the determination of what to store in the instruction and data caches of the Morrison system are made during execution of the program. Specifically,

> The execution set hardware operates independently of instruction fetching to control the movement of instruction words from main memory to the instruction cache. This hardware is responsible for fetching basic blocks of instructions into the cache until either the entire execution set resides in cache or program execution has reached a point that a branch has occurred to a basic block outside the execution set.

[Col. 24, lines 53-60]. Thus, when a processor requests an instruction from main memory, the hardware transfers to the instruction cache the entire set of instructions in that same execution set. As explained above, the execution set contains the re-ordered instructions. The decision to cache instructions has nothing to do with cacheability determinations made during compilation. Instead, all of the instructions are cached whenever the execution set in which they are contained is executed.

Data is stored in the data cache of the Morrison *et al.* system in essentially the same manner that instructions are stored in the instruction cache. Specifically, when a processing element attempts to read data, "[i]f the requested datum is not present in the data cache, the address delivered to the cache 1592 is sent to the data cache ATU 1580 to be translated into a system address. The system address is then issued to memory. In response, a block of data from memory (a cache line or block) is delivered into the cache partition circuits 1592 under control of data cache control 1586." [Col. 28, lines 50-56].

The Morrison *et al.* patent thus fails to disclose the subject matter of claim 1 that is missing from the teachings of the Mattson, Jr. *et al.* patent, *i.e.*, making cacheability determinations during compilation and using such cacheability determinations as a basis for determining which portions of data or instructions stored in the main memory should be stored in cache memory.

C. *The Teaching Of The Mattson, Jr. et al. Patent Cannot Properly Be Combined With The Teachings Of The Morrison et al. Patent*

It is well settled that it is not proper to combine the teachings of two or more references to support an obviousness rejection in the absence of some teaching in either reference of the desirability of combining their respective teachings. The Office Action does not cite any portion of either patent that provides this teaching. Moreover, it does not seem possible for the

respective teachings of the references to be combined even in hindsight since they each use distinctly different techniques to more efficiently execute computer programs, *i.e.*, the Mattson, Jr. *et al.* system executes instructions using branch prediction and the Morrison *et al.* system executes instruction that have been re-ordered. In fact, the Morrison *et al.* patent specifically teaches *not* caching instructions when a branch instruction is encountered (See, above quote, basic blocks of instructions a fetched into the cache "until ... a branch has occurred").

In summary, the teachings of the cited references cannot properly be combined, but even if they were combined, they still would not teach a computer system that stores in cache memory portions of the information stored in the main memory based on cacheability determinations that were made during compilation of a computer program being executed by the computer system. The rejection of claim 1 should therefore be withdrawn. The rejections of the remaining independent claims as being anticipated by the patent to Mattson, Jr. *et al.* are also improper and should be withdrawn.

Respectfully submitted,
DORSEY & WHITNEY LLP

Edward W. Bulchis
Registration No. 26,847
Telephone No. (206) 903-8785

EWB:pep

Enclosures:
Fax Cover Sheet
Fee Transmittal Sheet (+ copy)

DORSEY & WHITNEY LLP
1420 Fifth Avenue, Suite 3400
Seattle, Washington 98101-4010
(206) 903-8800 (telephone)
(206) 903-8820 (fax)

g:\clients\micron technology\00\500050.01\500050.01 req reconsideration 092205.doc